SQL

CREATE TABLE post\_ratings (

id INT(11) NOT NULL AUTO\_INCREMENT,

post\_id INT(11) NOT NULL,

user\_id INT(11) NOT NULL,

rating ENUM('+', '-') NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (post\_id) REFERENCES posts(id)

);

CREATE TABLE post\_ratings (

id INT(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

user\_id INT(11) UNSIGNED NOT NULL,

post\_id INT(11) UNSIGNED NOT NULL,

rating INT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

CONSTRAINT fk\_user\_id FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,

CONSTRAINT fk\_post\_id FOREIGN KEY (post\_id) REFERENCES posts(id) ON DELETE CASCADE

);

PHP

class PostRating {

private $db;

public function \_\_construct($db) {

$this->db = $db;

}

public function addRating($postId, $userId, $rating) {

if ($this->hasRated($postId, $userId)) {

return false;

}

$stmt = $this->db->prepare("INSERT INTO post\_ratings (post\_id, user\_id, rating) VALUES (?, ?, ?)");

$stmt->bind\_param("iis", $postId, $userId, $rating);

$stmt->execute();

return true;

}

public function getRating($postId) {

$stmt = $this->db->prepare("SELECT SUM(IF(rating = '+', 1, -1)) AS rating\_sum FROM post\_ratings WHERE post\_id = ?");

$stmt->bind\_param("i", $postId);

$stmt->execute();

$result = $stmt->get\_result()->fetch\_assoc();

return $result['rating\_sum'];

}

public function hasRated($postId, $userId) {

$stmt = $this->db->prepare("SELECT COUNT(\*) AS count FROM post\_ratings WHERE post\_id = ? AND user\_id = ?");

$stmt->bind\_param("ii", $postId, $userId);

$stmt->execute();

$result = $stmt->get\_result()->fetch\_assoc();

return $result['count'] > 0;

}

}

// Utworzenie obiektu klasy PostRating i połączenie z bazą danych

$postRating = new PostRating($db);

// Dodanie oceny dla danego posta i użytkownika

$postId = 1;

$userId = 2;

$rating = '+';

$postRating->add

<?php

class PostRatingSystem {

private $db;

function \_\_construct($db) {

$this->db = $db;

}

// Zapisuje ocenę posta przez użytkownika

function ratePost($user\_id, $post\_id, $rating) {

$query = "SELECT \* FROM post\_ratings WHERE user\_id = $user\_id AND post\_id = $post\_id";

$result = $this->db->query($query);

if ($result && $result->num\_rows > 0) {

// Użytkownik już ocenił ten post wcześniej, więc nie pozwalamy mu na ponowne oddanie głosu

return false;

} else {

// Dodajemy nowy wpis do tabeli post\_ratings z informacją o tym, że użytkownik o ID $user\_id dał ocenę $rating postowi o ID $post\_id

$query = "INSERT INTO post\_ratings (user\_id, post\_id, rating) VALUES ($user\_id, $post\_id, $rating)";

$this->db->query($query);

return true;

}

}

// Zwraca sumę punktów dla danego posta (dodatnią lub ujemną)

function getPostRating($post\_id) {

$query = "SELECT SUM(rating) AS rating\_sum FROM post\_ratings WHERE post\_id = $post\_id";

$result = $this->db->query($query);

if ($result && $result->num\_rows > 0) {

$row = $result->fetch\_assoc();

return $row['rating\_sum'];

} else {

return 0;

}

}

}

// Model bazy danych dla tabeli post\_ratings

class PostRating {

public $id;

public $user\_id;

public $post\_id;

public $rating;

function \_\_construct($id, $user\_id, $post\_id, $rating) {

$this->id = $id;

$this->user\_id = $user\_id;

$this->post\_id = $post\_id;

$this->rating = $rating;

}

}

?>

Wyjaśnienie   
Ten kod przedstawia implementację systemu oceniania postów w języku PHP, który jest oparty na uwierzytelnianiu użytkowników.

W kodzie mamy dwie klasy - PostRatingSystem i PostRating.

Klasa PostRatingSystem zawiera funkcje do zapisywania ocen użytkowników dla postów, wyliczania sumy punktów dla danego posta i uniemożliwiania ponownego oddania głosu przez tego samego użytkownika na ten sam post.

Klasa PostRating natomiast przedstawia model danych dla tabeli w bazie danych, która przechowuje informacje o ocenach użytkowników dla postów.

W kodzie używamy zapytań do bazy danych w celu pobierania i zapisywania informacji o ocenach użytkowników dla postów.

Ogólnie rzecz biorąc, kod ten zapewnia prosty system oceniania postów wraz z logowaniem ocen użytkowników i uniemożliwia ponowne oddanie głosu przez tego samego użytkownika.